CSI 436/536
Introduction to Machine Learning

SVM algorithm

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York
Solving SVM: separable case

- SVM in separable case is to
  \[
  \text{Minimize } ||w||^2, \text{ subject to: } y_i(w \cdot x_i + b) \geq 1
  \]

- How do we solve this quadratic programming problem numerically?
constrained optimization

- to solve \( \min_x f(x) \) s.t., \( g(x) \leq 0 \)
  - general idea: convert to unconstrained problem
  - three types of general methods
    - the barrier method, e.g.,
      \( \min_x f(x) + \log(-g(x)) \): always feasible
    - the penalty method, e.g.,
      \( \min_x f(x) + \max(0,g(x)) \): can be infeasible
    - primal-dual method, using Lagrangian duality
constrained optimization

- Lagrangian and Lagrangian multipliers for the *primal* problem
  \[ \min_x f(x) \text{ s.t., } g(x) \leq 0 \]
  - introduce multiplier \( 0 \leq \lambda \) and form Lagrangian
    \[ L(x, \lambda) = f(x) + \lambda g(x) \]
  - for any feasible \( x \), \( L(x, \lambda) \leq f(x) \), i.e., a lower bound

- dual problem
  - first, find \( x^*(\lambda) = \arg\min_x L(x, \lambda) \)
  - dual function: \( h(\lambda) = L(x^*(\lambda), \lambda) \) is *concave*
  - \( \max_\lambda h(\lambda), \text{ s.t., } 0 \leq \lambda \) is the *dual problem*
weak & strong duality

- \( f^* = \) optimal value of the primal problem
  \[ \min_x f(x) \text{ s.t., } g(x) \leq 0 \]

- \( h^* = \) optimal value of the dual problem
  \[ \max_\lambda h(\lambda), \text{ s.t., } 0 \leq \lambda \]

- with very loose conditions, we always have
  \( h^* \leq f^* \)
  this is known as the weak duality

- with more assumptions (e.g., primal problem is convex), we have
  \( h^* = f^* \)
  this is known as the strong duality

- many problem can be solved easily in the dual form
KKT condition

- Karush-Kuhn-Tucker condition
  
  - gradient of Lagrangian has to be zero
    \[ \nabla f(x) + \lambda \nabla g(x) = 0 \]

- primal feasibility: \( g(x) \leq 0 \)

- dual feasibility: \( \lambda \geq 0 \)

- complementary slackness: \( \lambda g(x) = 0 \)

- counterpart of the optimal condition of \( \nabla f(x) = 0 \) for unconstrained optimization
understanding the KKT condition

• Two cases

• Case 1: optimal solution inside feasible region
  \( \nabla f(x) = 0, \lambda = 0, g(x) < 0 \)

• Case 2: optimal solution on boundary
  \( \nabla f(x) \propto -\nabla g(x), \lambda > 0, g(x) = 0 \)
Example

- $\min_{x,y} f(x,y) = x^2 + 2y^2$, s.t., $x + y \geq 1$

Two constraints
1. Parallel normal constraint (= gradient constraint on $f, g$; solution is a max)
2. $G(x) = 0$ (solution is on the constraint line)

We now recast these by combining $f, g$ as the Lagrangian
understanding the KKT condition

• optimal solution
  • inside the feasible region
    • gradient of objective function is zero
  • on the boundary of the feasible region
    • gradient of objective function is orthogonal to the linear constraint form the boundary
• which case is indicated by the Lagrangian multiplier $\lambda \geq 0$
  • $\lambda = 0$: inside feasible region
  • $\lambda > 0$: on the boundary of feasible region
solving SVM: separable case

Primary problem

\[
\min_{\mathbf{w}} \quad \frac{1}{2} \| \mathbf{w} \|^2 \\
\text{s.t.} \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, \ldots, n
\]

Introducing multipliers \( \alpha_i \geq 0 \) and forming Lagrangian

\[
L(\mathbf{w}, b, \alpha) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{n} \alpha_i y_i (\mathbf{w}^T \mathbf{x}_i + b) + \sum_{i=1}^{n} \alpha_i.
\]
solving SVM: separable case

• We can solve the primary problem directly
  • Solution always exist when data are separable
  • But some elegant geometry is buried in the solution
• We instead solve the dual problem after removing primal variables because
  • KKT condition requires many multipliers to take zero values
  • training examples whose corresponding multiplier take nonzero values are the support vectors
solving SVM: separable case

Eliminate primal variables $w$ and $b$

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$$

From the first condition, we have $w = \sum_{i=1}^{n} \alpha_i y_i x_i$.

From the second condition, we have $\sum_{i=1}^{n} \alpha_i y_i = 0$.

Complementary slackness (from KKT condition)

$$\alpha_i (y_i (w^T x_i + b) - 1) = 0.$$
solving SVM: separable case

Eliminate primal variables \( w \) and \( b \) with \( w = \sum_{i=1}^{n} \alpha_i y_i x_i \) and \( \sum_{i=1}^{n} \alpha_i y_i = 0 \), the dual problem becomes

\[
\begin{align*}
\max_{\alpha} & \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{s.t.} & \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0.
\end{align*}
\]
Support vectors

Moving a support vector moves the decision boundary

Moving the other vectors has no effect
solving SVM: non-separable case

Minimize:

\[ \|w\|^2 + C \sum_{i=1}^{m} \xi_i \]

subject to:

\[ y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \]

Dual form:

\[ \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \]

s.t. \[ \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \]
Solving SVM

• The quadratic programming problem for either separable and non-separable cases can be solve efficiently using off-the-shelf packages

• We introduce however a particularly simple optimization scheme known as sequential minimization optimization (SMO) based on the paper of John Platt in 1996

  • This is the SVM algorithm I implemented in C

• Idea: coordinate descent
SMO for SVM

\[
\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

s.t. \( \sum_{i=1}^{n} \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq C \)

- Coordinate ascent: updating each element individually to reduce the optimization problem to a sequence of low-dim optimization problems
- however, for SVM, this does not work [Why?]
SMO for SVM

- each time optimize w.r.t. a pair of variables and reduce the problem to

\[
\max_{\alpha} \ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.
\]

s.t. \( 0 \leq \alpha_i \leq C, \ i = 1, \ldots, m \)

\[
\sum_{i=1}^{m} \alpha_i y^{(i)} = 0.
\]

\[
\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = - \sum_{i=3}^{m} \alpha_i y^{(i)}.
\]

\[
\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \zeta. \quad \alpha_1 = (\zeta - \alpha_2 y^{(2)}) y^{(1)}.
\]

\[W(\alpha_1, \alpha_2, \ldots, \alpha_m) = W((\zeta - \alpha_2 y^{(2)}) y^{(1)}, \alpha_2, \ldots, \alpha_m)\]
SMO for SVM

- Each time minimize a simple quadratic function with two variables and box constraints

\[ W(\alpha_1, \alpha_2, \ldots, \alpha_m) = W((\zeta - \alpha_2 y^{(2)}) y^{(1)}, \alpha_2, \ldots, \alpha_m) \]

\( y_1 \neq y_2 \Rightarrow \alpha_1 - \alpha_2 = k \)

\( y_1 = y_2 \Rightarrow \alpha_1 + \alpha_2 = k \)
SMO for SVM

Repeat till convergence {

1. Select some pair $\alpha_i$ and $\alpha_j$ to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).

2. Reoptimize $W(\alpha)$ with respect to $\alpha_i$ and $\alpha_j$, while holding all the other $\alpha_k$’s ($k \neq i, j$) fixed.

}

SVM solvers

- Many SVM solvers for python and other languages
  - Scikit-learn
  - LibSVM
  - SVM-light
  - SVM-torch
  - Matlab ML toolkit